Amendments to the Specification:

Please replace paragraph [0009] with the following replacement paragraph [0009]:

[0009] Other techniques for propagating address space employ a hit list. Using such ~~that~~ techniques, a block of address space is forwarded to a computing device, which then divides the hit list and passes portions on to found computing devices. This allows a complete ~~completely~~ randomization of the address space, but also requires a large file to be transmitted over the network.

Please replace paragraph [0014] with the following replacement paragraph [0014]:

[0014] Preferably, the mapping is a function based on a primitive element. ~~The network~~

Please replace paragraph [0029] with the following replacement paragraph [0029]:

[0029] Figure 2 further illustrates communication among computing devices by way of communication network 160. In this instance, computing device Foo.com desires [[to]] communication with Bar.com. In a conventional communication, Foo.com may have some means of translating the named address "Bar.com" to its IP address (i.e. 209.92.56.2). Of course, foo.com and bar.com are used here for convenience. They will in practice be referred to according to their IP addresses. Additionally, Foo.com may want to communicate directly with Bar.com by using its IP address. In fact, some computing devices that act as clients may not have a "name" address because they are not providing services to the outside world that requires communication by way of a domain name. Instead, such systems initiate the communication and provide their IP addresses ~~address~~ as part of the handshaking with the system with which they are ~~it is~~ communicating. In the case of Internet propagation systems, wherein the object is to communicate with each computing device connected to a given network, the domain name is of less significance.

Please replace paragraph [0033] with the following replacement paragraph [0033]:

[0033] To further illustrate aspects of the invention, Figure 3A illustrates a scanning technique whereby software may be propagated through a network using a

technique akin to propagating an Internet worm. Here, the Internet address space is divided into ~~in to~~ two portions. For example, the first portion 31 ranges from 0 to 2^{15}. The second portion 32 ranges from 2^{16} to $2^{32} - 1$. Of course, any other suitable division could be used. Each range is preferably assigned to a separate computing device, e.g., portion 31 is assigned to computing device 10a and portion 32 is assigned to computing device 10c. Of course, the initial address space could be further subdivided to any number of computing devices to start the process. However, if the goal is to propagate over the entire Internet ~~internet~~ space, a handful of additional computing devices will not substantially accelerate the propagation.

Please replace paragraph [0034] with the following replacement paragraph [0034]:

[0034] Each computing device 10a, 10c, then begins scanning through the range of addresses within its portion 31, 32, respectively, in the manner briefly described with respect to Figure 2. To that end, computing device 10a scans its assigned portion 31 attempting to find two computing devices ~~that~~ to which it can deliver a payload (e.g. payload 62 in Figure 2). In this example, computing device 10a scans portion 31a (shown with cross hatching) before finding and delivering a payload (e.g., payload 62) to two computing devices. Thereafter, the payload having been delivered, the newly enlisted computing devices execute the payload. As part of the operation of the payload, the newly enlisted computing devices each repeat the process of attempting to find two computing devices to which they can in turn deliver a copy of the payload, e.g., payload 62.

Please replace paragraph [0038] with the following replacement paragraph [0038]:

[0038] This process of finding two computing devices, providing them with the payload and subdividing the address space is also being performed on address space 32 as it splits into address spaces 321 and 322 and splits again into address spaces 324 ~~325~~ and 325 and address spaces 326 and 328. Address spaces 324, 325, 326, and 328 again split into address spaces 340, 342, 344, and 346. At each split, the previously scanned portions (e.g. 31a and 32a) do not need to be rescanned and are typically excluded from the portion that is subdivided among the newly enlisted computing devices.

Please replace paragraph [0041] with the following replacement paragraph [0041]:

[0041] Figure 4 illustrates a further aspect of the present invention. As shown, a function *f*(P) 41 is provided that maps the IP address space 30 to address space 30'. The characteristics of function *f*(P) 41 are such ~~is~~ that it provides a one-to-one mapping from address space 30 to address space 30'. Moreover, for every address in address space 30', there is a corresponding address in address space 30 and vice versa [[vice-versa]]. Hence, function *f*(P) 41 essentially shuffles the addresses in address space 30 into address space 30'. One ~~On~~ such function that can be used to perform this transformation makes use of a primitive element (see generator in [Neal Koblitz, "A Course in Number Theory and Cryptography", Second Edition, Graduate Texts in Mathematics, Springer, 1994]) to index IP addresses. The result is to randomly map an IP address *i* in address space ~~group~~ 30 to an element *e* in address space ~~group~~ 30' using a primitive element *p*.

Please replace paragraph [0042] with the following replacement paragraph [0042]:

[0042] Pseudo-randomness is achieved by traversing through the IP range as powers of a primitive element. Additional true randomness is achieved by the randomness of which computers are exploitable (the ~~The~~ splitting of IP ranges is completely dependent ~~dependant~~ on this).

Please replace paragraph [0043] with the following replacement paragraph [0043]:

[0043] Exponential propagation ~~Exponentially~~ is achieved by splitting the work load in half at each step. The Internet ~~internet~~ has 2^{32} addresses. The algorithm scans every computing device on the Internet ~~internet~~ in 32 steps.

Please replace paragraph [0044] with the following replacement paragraph [0044]:

[0044] One-to-One mapping is achieved by the definition of a primitive element. As a result, every host on the Internet ~~internet~~ with a valid IP is scanned once and only once.

Please replace paragraph [0046] with the following replacement paragraph [0046]:

[0046] After applying the above algorithm to the IP address space, Figure 5A illustrates a technique whereby software may be propagated through a network using a

technique akin to propagating an Internet worm. Here, the Internet address space is divided into ~~in to~~ two portions. For example, the first portion 31' ranges from P^0 to $P^{2\wedge 15}$; that set of addresses ~~address~~ was produced by setting Lowerbound to 0 and Upperbound to 2^{15} in the above algorithm. The second portion 32' ranges from $P^{2\wedge 16}$ to $P^{2\wedge 32-1}$ and results from setting Lowerbound to 2^{16} and Upperbound to $2^{32} - 1$ in the above algorithm. Of course, any other suitable division could be used. Each range is preferably assigned to a separate computing device, e.g., portion 31' is assigned to computing device 10a and portion 32' is assigned to computing device 10c. Of course, the initial address space could be further subdivided to any number of computing devices to start the process. However, if the goal is to propagate over the entire Internet ~~internet~~ space, a handful of additional computing devices will not substantially accelerate the propagation. The use of the binary tree is O(log(n)), where n is the size of the space, complete and would linearly improve by addition of child nodes.

Please replace paragraph [0047] with the following replacement paragraph [0047]:

[0047] Each computing device 10a, 10c, then begins traversing through the range of addresses within its portion 31', 32', respectively, in the manner briefly described with respect to Figure 2. To that end, computing device 10a traverses its assigned portion 31' attempting to find two computing devices ~~that~~ to which it can deliver a payload (e.g., payload 62 in Figure 2). Notably, unlike the scanning associated with Figures 3A and 3B, this technique will cause requests ~~request~~ to various computing devices to appear random in time and sequence.

Please replace paragraph [0049] with the following replacement paragraph [0049]:

[0049] One application of the above propagation technique would be to provide a patch or software update to software systems on a network including the Internet or a subnet. In that instance, the randomness of the propagation would relieve the traffic on any one ~~on~~ portion of a large network. That is, a large software update that went through a single network point in sequential fashion would tend to overwhelm and slow that portion of the network. Hence, a seemingly random process would tend to spread the load across the entire network.